Marker:

Programming language features in implemented and tested in the assignment		
Y/N	**Marks**	
0	0	class declarations
0	0	data member declarations
0	0	member function declarations
0	0	inheritance list
0	0	private/public members
0	0	free function definitions
0	0	member function definitions
0	0	variable declaration block
0	0	int, float, variable declarations
0	0	array variable declarations
0	0	if statement
0	0	while statement
0	0	read/write statement
0	0	return statement
0	0	assignment statement
0	0	complex direct structures involving [] and . and expressions as array indexes
0	0	complex expressions involving add(op, multiop, relop, unaryop, or/and), function calls

Notes	Remark (A)	ratio (on 100%)	Y/N		
	0%	10.00%		1	**Document Section 1 - Attribute grammar**
	0%	0.00%	X	1.1	Clarity - presentation of grammar and the problems/transformations
	0%	0.00%	X	1.2	Completeness - all problems with the grammar are listed/explained
	0%	0.00%	X	1.3	Statement of changed specification versus original specification; in the assignment statement (if applicable)
	0%	10.00%		2	**Document Section 2 - Description/rationale of the overall structure of the solution and the roles of the individual components used in the applied solution to the stated problem**
	0%	0.00%	X	2.1	Clarity - presentation of the design is clear and understandable
	0%	0.00%	X	2.2	Completeness - all implemented components are mentioned in the design description
	0%	0.00%		3	**Document Section 3 - Description of tools/libraries/techniques used in the analysis/implementation. Description of other tools that might have been used. Justification of why the chosen tools were selected**
	0%	0.00%	X	3.1	Clarity - the lexer/parser/implementation are mentioned
	0%	0.00%	X	3.2	Justification for lib tools
	0%	60.00%		4	**Correct implementation of syntax directed translation implemented as part of a top-down predictive parser following the grammar given in this handout, which is generating and AST (not a parse tree) data structure that correctly represents any valid program given in input;**
	0%	0.00%	X	4.1	class declarations
	0%	0.00%	X	4.2	data member declarations
	0%	0.00%	X	4.3	member function declarations
	0%	0.00%	X	4.4	inheritance list
	0%	0.00%	X	4.5	private/public members
	0%	0.00%	X	4.6	free function definitions
	0%	0.00%	X	4.7	member function definitions
	0%	0.00%	X	4.8	variable declaration block
	0%	0.00%	X	4.9	int, float, variable declarations
	0%	0.00%	X	4.10	array variable declarations
	0%	0.00%	X	4.11	if statement
	0%	0.00%	X	4.12	while statement
	0%	0.00%	X	4.13	read/write statement
	0%	0.00%	X	4.14	return statement
	0%	0.00%	X	4.15	assignment statement
	0%	0.00%	X	4.16	complex direct structures involving [] and . and expressions as array indexes
	0%	0.00%	X	4.17	complex expressions involving add(op, multiop, relop, unaryop, or/and), function calls
	0%	10.00%		5	**Output - output file: Output of clearly readable AST in a file either in a text file or in dot format file**
	0%	0.00%	X	5.1	Clarity - file is clearly readable if it is a text file or is successfully readable by a dot reader to render the AST as a graph
	0%	0.00%	X	5.2	Correctness - the graph is an AST as opposed to a parse tree
	0%	0.00%	X	5.3	Completeness - all syntactical constructs' representations are output to a file
	0%	10.00%		6	**Test cases - completeness of testing (in addition to the provided files)**
	0%	0.00%	X	6.1	class declarations
	0%	0.00%	X	6.2	data member declarations
	0%	0.00%	X	6.3	member function declarations
	0%	0.00%	X	6.4	inheritance list
	0%	0.00%	X	6.5	private/public members
	0%	0.00%	X	6.6	free function definitions
	0%	0.00%	X	6.7	member function definitions
	0%	0.00%	X	6.8	variable declaration block
	0%	0.00%	X	6.9	int, float, variable declarations
	0%	0.00%	X	6.10	array variable declarations
	0%	0.00%	X	6.11	if statement
	0%	0.00%	X	6.12	while statement
	0%	0.00%	X	6.13	read/write statement
	0%	0.00%	X	6.14	return statement
	0%	0.00%	X	6.15	assignment statement
	0%	0.00%	X	6.16	complex direct structures involving [] and . and expressions as array indexes
	0%	0.00%	X	6.17	complex expressions involving add(op, multiop, relop, unaryop, or/and), function calls
	0%	0.00%		7	**Successful/correct use of tools/libraries/techniques in the analysis/implementation.**
	0%	0.00%	X	7.1	The program never crashes or throws exceptions
	0%	0.00%	X	7.2	Tools presented in the labs are used appropriately, or comparable tools are used appropriately

Marker instructions
Enter values only in the red cells. Everything else is calculated automatically.
In Column D, enter either A, B, C, or F for each marking element
Enter notes in column A. These notes will be then automatically uploaded on moodle as feedback to the students.

Designer instructions
Add/remove some lines with red cells to add/remove evaluation criteria
When you add lines, add them only in the middle of the area, or else the calculation is going to be wrong
To add lines, add a new row, then copy into it one of the existing rows.

Marking mark	Ratio		Raw ratio	Total marks
Document Section 1 - Attribute grammar	0	0.00	0.00	0.00
Document Section 2 - Description/rationale of the overall structure of the solution and the roles of t	0	0.00	0.00	0.00
Document Section 3 - Description of tools/libraries/techniques used in the analysis/implementation.	0	0.00	0.00	0.00
Correct implementation of syntax directed translation implemented as part of a top-down predictive	0	0.00	0.00	0.00
Output - output file: Output of clearly readable AST in a file either in a text file or in dot format fil	0	0.00	0.00	0.00
Test cases - completeness of testing (in addition to the provided files)	0.00	0.00	0.00	0.00
Successful/correct use of tools/libraries/techniques in the analysis/implementation.	0	0.00	0.00	0.00
Total	60	100.00		0.00